

## IN THE SPECIFICATION

*Please make the following amendments to the specification.*

Please replace paragraph 002 with the following paragraph.

[002] A well known technique ~~that solves this disadvantage of~~ used by conventional compilation systems is interprocedural analysis (IPA). IPA is a phase that is added to a compilation system to analyze an entire program and collect global information related to the translation units. Global information includes global variables and how the multiple translation units manipulate and reference the global variables. Once the global information is collected, it is then passed to the optimizer as part of the back end of the compilation system. Thus, when the optimizer optimizes a translation unit, the optimizer accesses this global information and performs additional and more aggressive optimization pertaining to global variables. IPA improves the efficiency of the generated object code by providing optimization at a global level, thereby improving the run-time performance of the executable program.

Please replace paragraph 007 with the following paragraph.

[007] A “side-effect” may be a reading ~~of~~ or writing of storage by a routine. The method further comprises determining for each pointer parameter ~~to~~ of a routine, whether that routine reads and/or writes storage via the parameter or a pointer derived from it. For each routine, the pointer parameters that might be used to derive the routine’s return value are determined.

Please replace the abstract of the disclosure with the following paragraph.

[0078] ~~A method and system for Interprocedural side-effect analysis is disclosed~~  
performed by constructing a fixed-point problem graph for each translation unit of a ~~In one~~  
~~embodiment, the method is applied to a software program having a plurality of separately~~  
compilable components. The method performs analyzing each routine, of a software  
program having a plurality of separately compilable routines, to create a plurality of local  
side-effect problems for each routine; and merging the local side-effect problems to create a  
global side-effect problem.